METHOD FOR PRODUCING SOFTWARE MODULES FOR FIELD DEVICES OF PROCESS AUTOMATION TECHNOLOGY

The invention relates to a method for producing software modules for field devices of process automation technology.

In process automation technology, field devices are frequently used, which serve for registering and/or influencing process variables. Examples of such field devices are fill level measuring devices, mass flow meters, pressure meters, meters for measuring temperature, etc., which register the corresponding process variables fill level, mass flow rate, pressure, and temperature. Serving for the influencing of process variables are so-called actuators, which, e.g. as valves, influence the flow of a liquid in a section of pipeline.

The field devices are usually connected to a data bus, and are, as a rule, connected with a central control-, or engineering-, system, which controls the entire process flow and/or enables a direct access to the individual field devices. In the control system, the measured values of the different process variables are evaluated and/or monitored, and appropriate actuators are correspondingly activated for influencing the process. Data transmission between field device and control occurs according to the known international standards for field busses, such as e.g. Hart, Foundation Fieldbus, Profibus, CAM, etc.

Today's automated plants frequently involve a large number of different field devices of a widest variety of manufacturers. Before startup and/or during operation, adjustments must occur at the field devices. These adjustments must frequently be made on site. To this end, the individual field device manufacturers provide different configuration programs for the different devices. Becoming competent with the different programs, including the different operating philosophies, demands extreme effort and time on the part of the user.

The parametering of individual field devices or the configuring of certain groups of field devices in an automated plant having a multiplicity of field devices is extremely complex and expensive, because of the various communications interfaces and the required protocols. The configuring, operating, and maintaining of a field device in an automated plant should be considerably simpler to perform. Desired is the integration of field devices into control systems or engineering applications via a Plug and Play capability, such as is already known e.g. for printers in Windows environments.

Various field device manufacturers have, therefore, come together in PNO (PROFIBUS Nutzerorganisation e.V.), for the purpose of enabling a simpler handling of field devices. The field device manufacturers develop special software modules for their field devices. These software modules are delivered with the field devices to the customers. Each software module encapsulates all data and functions of the particular field device and represents, in principle, a black box. Additionally, the device manufacturer can integrate into these software modules its own "look and feel". I.e., the user interface always looks the same to the user, independently of the particular application. The application program, which serves e.g. for configuring, visualizing, operating and maintaining the different field devices, accesses the particular software modules of the field device via a defined interface. One possibility is the FDT/DTM interface specification, as given in the Profibus Guideline - Order No. 2.162 of November 2000, which can be obtained via the PNO, Karlsruhe, Germany (www.profibus.com). The content of such guideline is incorporated here by reference.

At the moment, corresponding software modules are available only for a few field devices. For a large number of field devices, the software modules still have to be produced by the pertinent manufacturers. One possibility is to convert available device descriptions by means of compilers, or generators, into corresponding software modules.

The available device descriptions, however, do not exist in a uniform form, or language. There are PDM device descriptions, HCF device descriptions, as well as company-specific device descriptions stored in internal company databases. For each of these categories of different device descriptions, a separate compiler is necessary.

Disadvantageous in the case of such method is that one must generate different compilers for different categories of device descriptions. A further disadvantage of such method is that, in the case of changes, all utilized compilers always have to be revised, in order to prevent inconsistencies. This makes the generating of software modules for field devices by means of present device descriptions very complex.

It is an object of the invention to provide a method for producing software modules for field devices of process automation technology, wherein the desired software modules are produced in simple and cost-favorable manner from existing device descriptions.

This object is achieved by the method steps given in claim 1.

An essential idea of the invention is not to apply different compilers for different groups of existing device descriptions, but, instead, to produce, from standard device descriptions for field devices, syntactically and semantically correct, standard device descriptions and then to convert these by means of a compiler into the corresponding software modules.

Advantageous further developments of the invention are set forth in the dependent claims.

The invention will now be explained in greater detail on the basis of an example illustrated in the drawing, the figures of which show as follows:

Fig. 1 schematic diagram of an installation for automation; and

Fig. 2 schematic illustration of essential elements of the method of the invention.

Fig. 1 is a schematic diagram of an installation for automation. A control system L is connected via a data bus D with a plurality of field devices F1, F2, F3, etc.. The field devices F1-F3 can be e.g. pressure meters, temperature meters, or flow meters. The control system L communicates via the data bus D with the relevant field device, e.g. F1. In this way, measured values or parameters of the field device can be transmitted to the control system L. At the same time, the parametering of the field device F1 can occur from the control system L.

Data communication on the data bus D occurs according to corresponding international standards, such as e.g. HART, Profibus, FF or CAN.

For currently marketed field devices, there exist many groups of device descriptions in different forms and languages. Examples are, as given in Fig. 2, PDM device descriptions, HCF device descriptions and a database DB containing company-internal device descriptions. At least the PDM device descriptions, as well as the HCF device descriptions, contain ambiguities.

In a first method step, syntactically and semantically correct, standard device descriptions are created from standard device descriptions (PDM device descriptions, HCF device descriptions, firm-specific device descriptions). Used for this purpose is the generator G1, or the compiler C1, as the case may be. Serving as an example of a semantically and syntactically correct, standard device description is EDD 1.1.

Then, the semantically and syntactically correct, standard device description is converted with the help of a compiler C into a software module SM (e.g. DTM), which has defined interfaces to application programs in the process control system. By producing the software module SM by way of the intermediate step through an EDD, the essential advantages of the method of the invention are obtained. Only a single compiler C is needed to produce the software module SM.

Furthermore, from the standard device descriptions (EDD 1.1) then available in syntactically and semantically correct form, PDM device descriptions can likewise be easily produced.

Beyond this, from device descriptions in EDD 1.1, a generator G2 can be used to produce graphical user interfaces in XML language GUI.XML for the corresponding, field-device-specific components (configuration, help-functions, etc.). For this, also the generator G1 can be used, for processing information from the database DB. The components produced in this way can then likewise be processed in the compiler C.

A further opportunity is to produce general graphical user interfaces GUIs and to store these in a library B (DTM-Studio component library). These components can likewise be linked in the compiler C, so that the software module SM also has proprietary interfaces for GUIs, or for components EC, or Lin., as the case may be. In this way, an envelope curve presentation capability (component EC), or a linearizing capability (component Lin.), as the case may be, can be integrated in simple manner into corresponding software modules SM.